Received: 7 May 2020 | Revised: 14 March 2021 | Accepted: 30 March 2021

DOI: 10.1111/sjos.12532

ORIGINAL ARTICLE

Scandinavian Journal of Statistics

An n-dimensional Rosenbrock distribution for Markov chain Monte Carlo testing

Filippo Pagani[1] | **Martin Wiegand**[1] | **Saralees Nadarajah**[2]

[1]MRC Biostatistics Unit, University of Cambridge, Cambridge, UK

[2]Department of Mathematics, University of Manchester, Manchester, UK

Correspondence
Filippo Pagani, MRC Biostatistics Unit, University of Cambridge, Cambridge CB2 0SR, UK.
Email: filippo.pagani@mrc-bsu.cam.ac.uk

Funding information
Engineering and Physical Sciences Research Council, Grant/Award Number: EP/R018561/1; Medical Research Council, Grant/Award Number: MC_UU_00002/10

Abstract

The Rosenbrock function is a ubiquitous benchmark problem in numerical optimization, and variants have been proposed to test the performance of Markov chain Monte Carlo algorithms on distributions with a curved and narrow shape. In this work we discuss the Rosenbrock distribution and the advantages and limitations of its current n-dimensional extensions. We then propose a new extension to arbitrary dimensions called the Hybrid Rosenbrock distribution, which addresses all the limitations that affect the current extensions. The Hybrid Rosenbrock distribution is composed of conditional normal kernels arranged in such a way that preserves the key features of the original Rosenbrock kernel. Moreover, due to its structure, the Hybrid Rosenbrock distribution is analytically tractable, and possesses several desirable properties which make it an excellent test model for computational algorithms. We conclude with numerical experiments that show how commonly used Markov chain Monte Carlo algorithms may fail to explore densities with curved correlation structure, restating the importance of a reliable benchmark problem for this class of densities.

KEYWORDS

algorithm testing, benchmarking, Markov chain Monte Carlo, MCMC, Rosenbrock

1 | INTRODUCTION

The Rosenbrock function (Rosenbrock, 1960) is a popular test problem in the optimization literature due to its challenging features: Its minimum is located at the bottom of a long and narrow parabolic valley. The original function can be turned into a probability density that maintains these features, and has been adopted by the Markov chain Monte Carlo (MCMC) community to serve as a benchmark problem when testing MCMC algorithms (e.g., Goodman & Weare, 2010).

One of the current frontiers of research in this field is developing algorithms (e.g., Girolami et al., 2011; Parno, 2014) that can sample efficiently from densities that have 2-*d* marginals with nonconstant or curved correlation structure (see, e.g., Figure 1). Such shapes make it difficult for MCMC algorithms to take large steps, increasing the autocorrelation time and decreasing the quality of the MCMC sample.

Distributions with curved correlation structures often arise when dealing with complex or hierarchical models, typically found in cosmology (e.g., The Dark Energy Survey Collaboration, 2017), epidemiology (e.g., House et al., 2016), chemistry (e.g., Cotter et al., 2019), finance (e.g., Kim et al., 1998), biology (e.g., Christensen et al., 2005; Sullivan et al., 2010), ecology (e.g., Rockwood, 2015), particle physics (e.g., Allanach & Lester, 2008; Feroz et al., 2009), and many other fields. Sometimes reparametrizing the model can map the problematic components to more linear shapes, but it is not always possible, and the reparametrization may not solve the problem entirely.

Researchers developing new methods for distributions with nonlinear correlation structure often test their algorithms on only a handful of benchmark models, among which the (two-dimensional) Rosenbrock kernel is quite popular (Hogg & Foreman-Mackey, 2018). However, few properties of the Rosenbrock kernel have been investigated and formalized, especially regarding multivariate extensions of the density for the purpose of MCMC sampling. As we will show in Section 2.1, sometimes the properties of this distribution are so poorly understood that extending the kernel from two to three dimensions radically changes its shape.

In this work we present the Hybrid Rosenbrock distribution, a benchmark model that i) has curved 2-*d* marginal distributions; ii) is easily extendable to more than two dimensions; iii) has known normalization constant; iv) explains clearly the effects that changes in the parameters have on its shape; v) allows for direct and efficient Monte Carlo sampling. To our knowledge, there is no benchmark model for distributions with curved correlation structure that possesses all of the above properties.

The Hybrid Rosenbrock distribution can be used by researchers developing new MCMC methods to test how algorithms perform on distributions with curved 2-*d* marginals. Moreover, the Hybrid Rosenbrock distribution can also be used by savvy MCMC practitioners to perform algorithm selection. The shape and features of the Hybrid Rosenbrock can be tweaked to match those of the model of interest, which would provide the practitioners with a tailor-made toy problem to test their algorithm of choice and assess how well it performs when compared with the true solution.

Furthermore, the Hybrid Rosenbrock distribution can be used to test the accuracy of algorithms that estimate the normalizing constant of a kernel (Gelfand & Smith, 1990; Satagopan et al., 2000). Prominent approaches include Chib (1995), DiCiccio et al. (1997), and Moral et al. (2006), among various other contributions. Due to the number of approaches suggested,

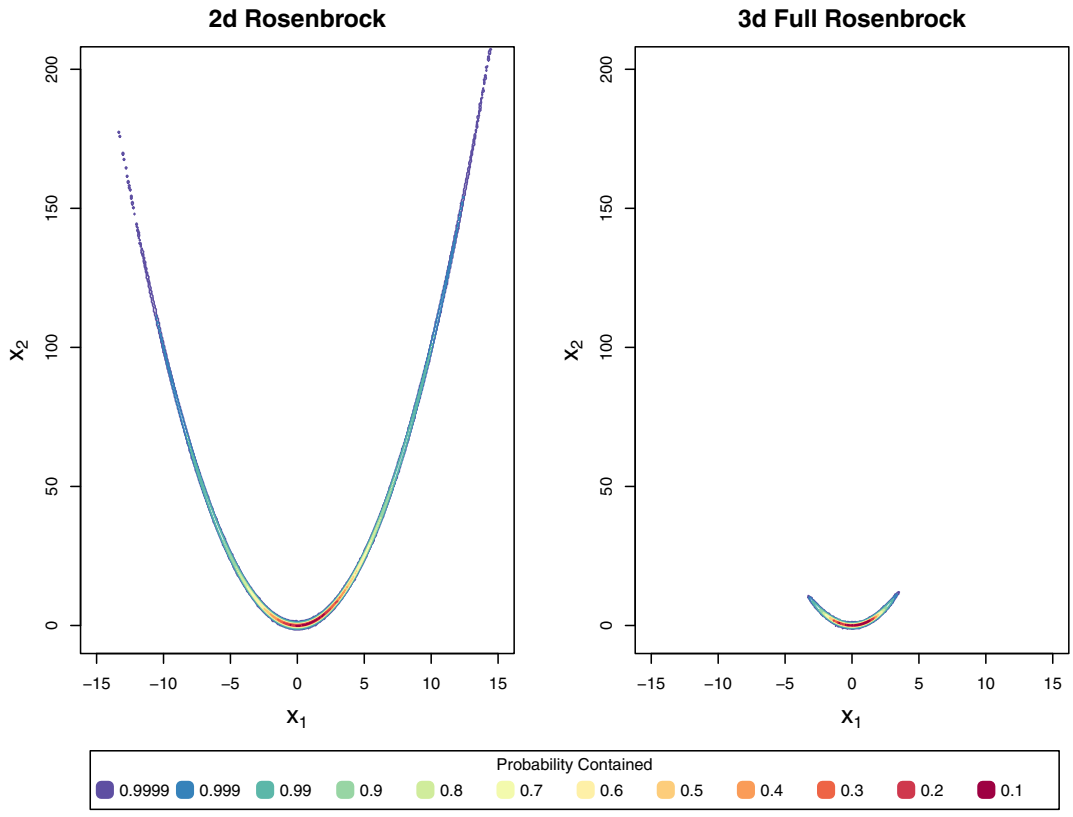

FIGURE 1 Contour plots of a 2-*d* Rosenbrock density as described in Equation (1), and of the x_1 and x_2 variables from a 3-*d* Full Rosenbrock kernel from Equation (2) [Color figure can be viewed at wileyonlinelibrary.com]

having a challenging benchmark problem for which the normalizing constant is known would prove a valuable assessment tool.

The structure of the article is as follows. In Section 2 we review the current literature on 2-*d* Rosenbrock distributions and the available *n*-dimensional extensions. In Section 3 we present our *n*-dimensional extension, and discuss how it improves on the shortcomings of current solutions. In Section 4 we discuss how changes in the structure and shape of the Hybrid Rosenbrock density affect the difficulty of obtain an MCMC sample from it, and compare the performance of some popular MCMC algorithms. In Section 5 we report our conclusions.

Our code is available in the form of a simple tutorial at https://github.com/FilippoPagani/hybridRosenbrock, and the R package "Rosenbrock" is available on CRAN.

2 | CURRENT LITERATURE

The simplest nontrivial case of the Rosenbrock distribution is the 2-*d* case, where the kernel can be written as

$$\pi(x_1, x_2) \propto \exp\left\{-[100\,(x_2 - x_1^2)^2 + (1 - x_1)^2]/20\right\}, \quad x_1, x_2 \in \mathbb{R}. \tag{1}$$

We follow (Goodman & Weare, 2010) when rescaling Equation (1) by 1/20, so that the distribution takes the shape of a curved narrow ridge—shown in Figure 1 on the left side—which is normally quite challenging for MCMC algorithms to explore.

It is not clear from the literature how the shape of the kernel in Equation (1) is affected by changes in the coefficients. Moreover, the normalizing constant is generally unknown, and there is more than one way to extend the distribution beyond two dimensions. Two methods have been proposed in the literature, and we will review them to point out their advantages and limitations.

2.1 | Full Rosenbrock distribution

We will refer to the n-dimensional extension in Goodman and Weare (2010) as "Full Rosenbrock" kernel in the following paragraphs. The kernel has the following structure:

$$\pi(\mathbf{x}) \propto \exp\left\{-\sum_{i=1}^{n-1}\left[100\,(x_{i+1}-x_i^2)^2+(1-x_i)^2\right]/20\right\}, \quad \mathbf{x}=[x_1,\,\dots\,,x_n]^\top \in \mathbb{R}^n. \tag{2}$$

The normalizing constant is unknown, and the effect of the coefficients on the shape of the distribution is unclear.

Figure 1 shows a comparison between the variables x_1 and x_2 from the 2-d kernel, and the same variables from a 3-d Full Rosenbrock kernel. A more exhaustive plot of a 3-d Full Rosenbrock kernel is shown in Figure A1 in Appendix A. The stark difference between the plots in Figure 1 shows how extending the Rosenbrock kernel from two to three dimensions as described in Equation (2) alters the joint plot between the variables x_1 and x_2. The long narrow ridge has become much more concentrated around the mode, decreasing the difficulty of the problem from an MCMC perspective.

However, the Full Rosenbrock kernel does have some desirable features: as n increases, the variance of x_n increases steeply. Densities with such properties (e.g., Neal's Normal in Neal, 2010) are challenging to MCMC algorithms, especially if the dependence between components is nonlinear.

2.2 | Even Rosenbrock distribution

In the optimization literature, Dixon and Mills (1994) proposes a simpler version of the Full Rosenbrock function used in Section 2.1, which can be turned into a kernel as

$$\pi(\mathbf{x}) \propto \exp\left\{-\sum_{i=1}^{n/2}\left[(x_{2i-1}-\mu_{2i-1})^2-100\,(x_{2i}-x_{2i-1}^2)^2\right]/20\right\}, \quad \mathbf{x}\in\mathbb{R}^n, \tag{3}$$

where n must be an even number, and we maintain the 1/20 mentioned in the previous section. The normalizing constant is unknown, and the effect of the coefficients on the shape of the distribution is unclear.

This density is in essence the product of $n/2$ independent blocks, each containing a 2-d Rosenbrock kernel. Unlike the Full Rosenbrock kernel, the Even Rosenbrock does maintain the shape of the joint 2-d marginals as n increases. However, only a small fraction of the joint distributions

are curved narrow ridges, while the majority of the 2-*d* marginals are uncorrelated (see Figure A2 in Appendix A for more details).

3 | THE HYBRID ROSENBROCK DISTRIBUTION

As outlined in Section 1, the overall goal of this article is presenting an n-dimensional benchmark model that has the required marginal structure, known normalization constant, parameters whose changes have clear effects on its shape, and it admits simple and robust Monte Carlo sampling. These properties are vital for a suitable benchmark distribution. The Hybrid Rosenbrock density fulfills all of the outlined properties, and draws on both models described in Section 2 to provide a reliable target where every single 2-*d* marginal distribution has a complex dependency structure. Its kernel can be written as:

$$\pi(\mathbf{x}) \propto \exp\left\{-a(x_1-\mu)^2 - \sum_{j=1}^{n_2}\sum_{i=2}^{n_1} b_{j,i}(x_{j,i} - x_{j,i-1}^2)^2\right\}, \tag{4}$$

where $\mu, x_{j,i} \in \mathbb{R}$; $a, b_{j,i} \in \mathbb{R}^+$ $(\forall j, i)$, and where the final dimension of the distribution is given by the formula $n = (n_1 - 1)n_2 + 1$. The dependency structure between the components $x_1, \ldots, x_{n_2,n_1}$ of the Hybrid Rosenbrock distribution can be represented with a graphical model as shown in Figure 2.

Each "row" of the diagram in Figure 2 represents a "block of variables." The index i in Equation (4) identifies variables within a "block", with n_1 denoting the block size, while the index j identifies a single block among the n_2 blocks present. The indices on the coefficients $b_{j,i}$ follow the block structure, as do the indices on the random variables $x_{j,i}$. The variable $x_{j,1} = x_1, \forall j = 1, \ldots, n_2$ is represented with only one index, as it is common to all blocks.

Figure 3 shows the contour plots obtained from a Monte Carlo sample of the kernel in Equation (4) when taking $n_2 = 2$, $n_1 = 3$, and $\mu = 1$, $a = 1/20$, and $b_{j,i} = 100/20$ $(\forall j, i)$, that is,

$$\begin{aligned}\pi(\mathbf{x}) \propto \exp\big\{&-a(x_1-\mu)^2 - b_{1,2}(x_{1,2} - x_1^2)^2 - b_{1,3}(x_{1,3} - x_{1,2}^2)^2 \\ &- b_{2,2}(x_{2,2} - x_1^2)^2 - b_{2,3}(x_{2,3} - x_{2,2}^2)^2\big\}, \quad \mathbf{x} \in \mathbb{R}^5.\end{aligned} \tag{5}$$

The Hybrid kernel inherits from the Full Rosenbrock kernel the feature of having variables with very different variances, as can be observed in the scales of the plots in Figure 3. Moreover, as opposed to the Even Rosenbrock (contours shown in Figure A2 in Appendix A), no plot in Figure 3 presents trivial correlation structure: every 2-*d* marginal is a straight or curved ridge with very long tails. At the same time, the Hybrid kernel inherits from the Even kernel the block

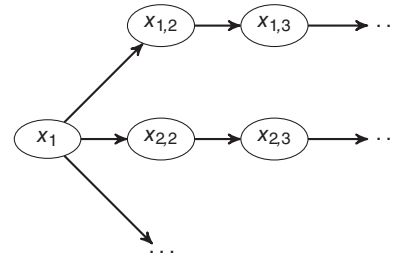

FIGURE 2 Graphical model representing the dependency structure of the Hybrid Rosenbrock distribution. The circles represent the kernels of each variable, while the edges represent the direct dependence between the kernels

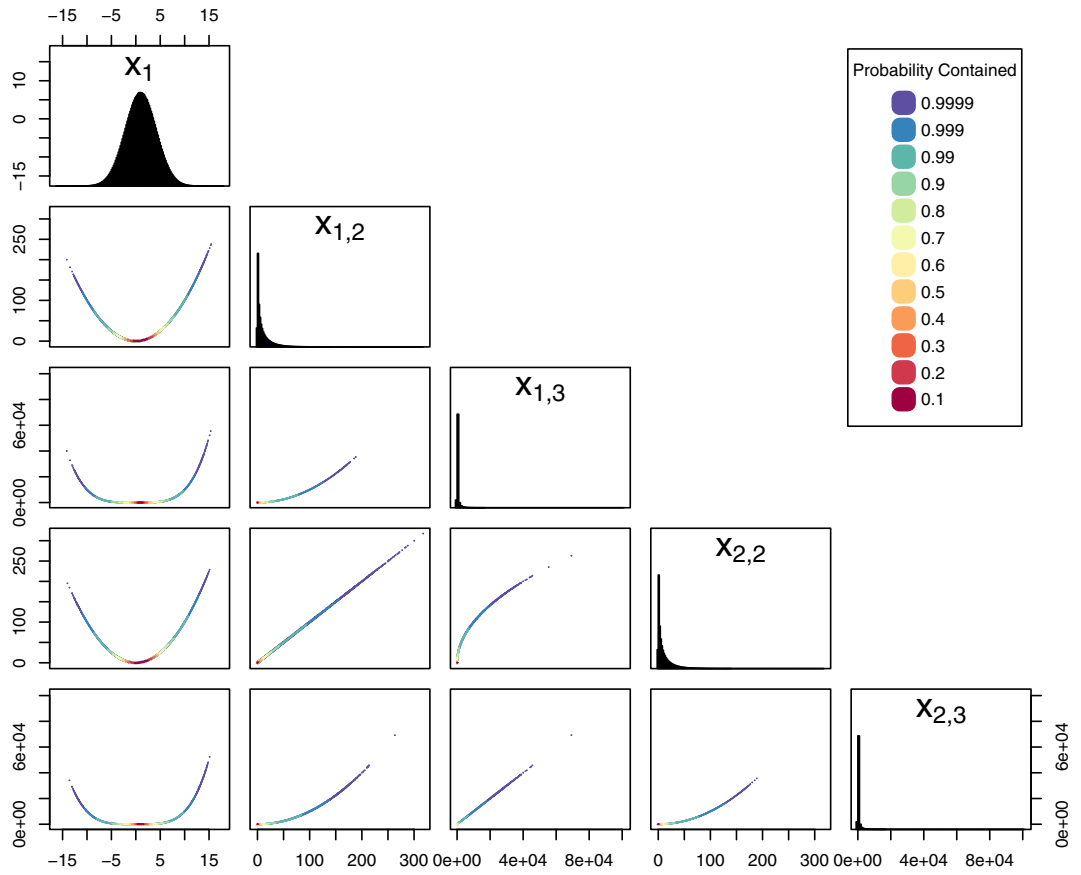

FIGURE 3 Contour plot of a $(n_1, n_2) = (3, 2)$ Hybrid Rosenbrock density as described in Equation (4), with parameters $a = 1/20$, $b_{j,i} = 100/20$ $(\forall j, i)$, $\mu = 1$, obtained via direct sampling. Every joint distribution is either a straight or curved ridge [Color figure can be viewed at wileyonlinelibrary.com]

structure, which guarantees that as n grows, the variance of each variable is computationally stable to compute. Remark 2 in Section 4 will address this point.

The true strength of the Hybrid Rosenbrock distribution lies not only in the way we connect the terms in Equation (4), but also in recognizing that each term in Equation (4) is in fact a conditional normal kernel. For example, the term

$$\exp\left\{-b_{j,i}(x_{j,i} - x_{j,i-1}^2)^2\right\}$$

from Equation (4) represents a conditional normal kernel $\mathcal{N}(x_{j,i}|x_{j,i-1}^2, (2b_{j,i})^{-1})$ with mean $x_{j,i-1}^2$ and variance $(2b_{j,i})^{-1}$, while the first term in Equation (4)

$$\exp\left\{-a(x_1 - \mu)^2\right\}$$

simply represents an unconditional normal kernel $\mathcal{N}(x_1|\mu, (2a)^{-1})$. This new perspective allows us to give the normalization constant for the Hybrid Rosenbrock distribution.

Proposition 1. *The normalization constant of the Hybrid Rosenbrock kernel given in Equation (4) with* $n = (n_1 - 1)n_2 + 1$ *is*

$$\frac{\sqrt{a}\prod_{i=2,j=1}^{n_1,n_2}\sqrt{b_{j,i}}}{\pi^{n/2}}.$$

Proof. We use the conditional structure of the density to split the integrals of the normal kernels and solve them one at a time. The details are shown in Appendix C. ∎

Interpreting the Hybrid Rosenbrock density as the composition of normal kernels also provides us with a simple interpretation for the coefficients. As $2a$ and $2b_{j,i}$ represent the precisions of the conditional normal kernels, increasing a increases the slope of the distribution *along* the ridge formed around the parabola $x_{j,i-1}^2$, while increasing $b_{j,i}$ decreases the dispersion *around* the parabola. The parameter μ determines the position of the mode of the variable x_1 along the parabola.

Remark 1. In this work we will only investigate the case where the kernels are normal and connected through the mean function, and where the mean function is the parabola x^2. Our choice is based on the historical importance of the Rosenbrock function, and on the fact that "banana"-like distributions are common in many fields of knowledge. Indeed, any polynomial can be used as mean function, as well as other functions such as $\exp(x)$, $\sin(x)$, $1/x$. In fact, any function $f(x) : \mathbb{R} \to E \subseteq \mathbb{R}$ that does not alter the behavior of the integrals in the proof of Proposition 1 is a viable candidate as mean function. Furthermore, as long as the same conditions are satisfied, kernels other than normal can be used. This should provide more than enough variety for the practitioner to adapt the Hybrid Rosenbrock to their own specific problem. Depending on the choice of the mean function, the block structure can be adopted to control the variance of those components that grow too quickly, as mentioned in Remark 2.

Using the conditional normal structure of the model, it is possible to obtain an *i.i.d.* Monte Carlo sample from the joint distribution by using Algorithm 1. Notably, Algorithm 1 is not a Gibbs sampler as the first kernel in x_1 is always independent of any other kernel.

Algorithm 1. Pseudocode to sample from a Hybrid Rosenbrock distribution

$\mu = 1$, $a = 1/20$ and $b = 100/20$ **for** $k = 1, \dots, N$ **do**
  $X_1 \sim \mathcal{N}\left(\mu, \frac{1}{2a}\right)$
  for $j = 1, \dots, n_2$ **do**
    for $i = 2, \dots, n_1$ **do**
      $X_{j,i} | X_{j,i-1} \sim \mathcal{N}\left(x_{j,i-1}^2, \frac{1}{2b_{ji}}\right)$
    end
  end
  $X_{(k)} = (X_{(1)}, X_{1,2}, \dots, X_{n_2,n_1})$
end
return $\left(X_{(1)}, \dots, X_{(N)}\right)$

4 | NUMERICAL TESTS

In this section we investigate how varying the parameters $n_1, n_2, \mu, a, b_{j,i}$ $(\forall j, i)$ influences the performance of MCMC algorithms sampling from the Hybrid Rosenbrock distribution.

4.1 | Model comparison

In this section we compare the integrated autocorrelation time τ calculated using MCMC samples from models with different sets of parameters. The integrated autocorrelation time roughly measures how many steps on average an MCMC algorithm needs in order to return a sample that is completely uncorrelated with the original position $\mathbf{x}$. Studying how τ varies for each model provides insights into how easy it is to sample from that model via MCMC (see, e.g., Goodman & Weare, 2010 and references therein).

To obtain the MCMC samples we rely on a simplified manifold MALA (sMMALA) algorithm (Girolami et al., 2011) with SoftAbs metric (Betancourt, 2013), tuned with $\alpha = 10^6$ and acceptance ratio set at roughly 50%. sMMALA is an algorithm particularly well suited for this class of densities, as it uses the local correlation structure of the target to make more ambitious moves (see Appendix D for an accurate description). As all our models are multidimensional, each MCMC sample yields a vector of n autocorrelation times τ_i, one for each component of the state space, where τ_i is defined as

$$\tau_i = 1 + 2\sum_{l=1}^{L} \mathrm{Cor}(y_i^0, y_i^l), \quad i = 1, \dots, n,$$

where $\mathbf{y}_i$ is the MCMC sample from the ith component of the state space, and L is an integer number representing the last lag where the sample autocorrelation is significantly different from zero. We then record only the highest autocorrelation time among all components:

$$\tau = \max_{i=1,\dots,n} \tau_i.$$

Naturally, the smaller the value of τ, the better the algorithm mixes. Assuming the autocorrelation in an MCMC sample is always nonnegative, an algorithm that generates *i.i.d.* samples achieves the smallest possible value of τ, that is, $\tau = 1$. The integrated autocorrelation time is closely related to the effective sample size (ESS), as $\mathrm{ESS} = N/\tau$, where N is the number of Monte Carlo samples available.

The integrated autocorrelated times τ_i are calculated here with initial sequence estimators (see, e.g., Christen & Fox, 2010; Geyer, 1992), and the results are reasonably consistent between different runs, as the uncertainty bars confirm in Figure 5. Our results are also consistent with estimating τ_i by fitting an autoregressive (AR) process to the time series (Thompson, 2010). Unlike Roberts and Rosenthal (2001) we deliberately did not divide τ_i by n, as it is of interest to us to capture the effect that an increase in n has on the difficulty of sampling from the target.

In this section we will test six separate distribution structures or models, indexed by the parameters n_1, n_2, and for each of them, we will vary the parameters $\mu, a, b_{j,i}$ $(\forall j, i)$ to change the model's shape. For simplicity, we will fix $b_{j,i} = b$, $\forall j, i$. The structure of the six models is represented in Figure 4.

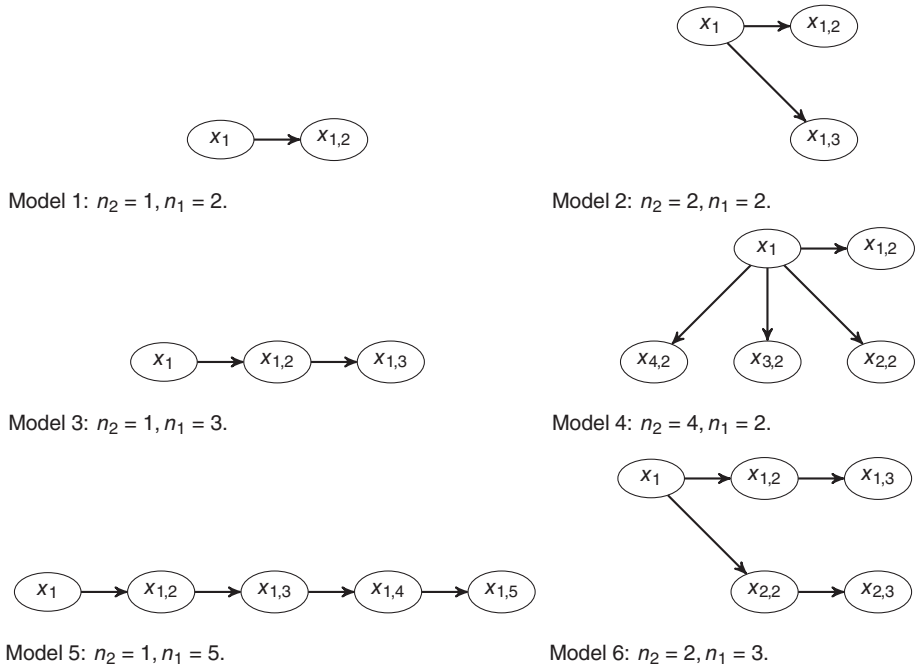

FIGURE 4 Graphical models of the six different Hybrid Rosenbrock structures tested in this section

Model 1 corresponds to the 2-*d* Rosenbrock density, that is, Equation (4) with $n_2 = 1$ and $n_1 = 2$, and is included to represent the baseline against which every other model is compared.

Model 2 ($n_2 = 2, n_1 = 2$) and Model 3 ($n_2 = 1, n_1 = 3$) are both 3-*d* distribution. Model 2 captures the effect of extending the 2-*d* density by adding an extra block, while Model 3 captures the effect of increasing the number of variables in the same block. We expect Model 3 to be more challenging than Model 2, as the difference between the variance of the variables in Model 3 should be higher than in Model 2.

Model 4 ($n_2 = 4$, $n_1 = 2$) and Model 5 ($n_2 = 1$, $n_1 = 5$) are simply larger versions of Models 2 and 3, and they capture the effects that an increase in dimension of the state space has on the sampling algorithm.

Overall, Models 2 to 5 are extensions of the 2-*d* case obtained by only increasing either the number of blocks or the number of variables in the single block available.

Instead, Model 6 ($n_2 = 2$, $n_1 = 3$) is a fully Hybrid Rosenbrock distribution, with multiple blocks and multiple variables in each block. Its 2-*d* marginals can be seen in Figure 3.

Remark 2. Model 5 was included for comparison, but it is a viable option only for certain parameter values and only in low dimension. The reason is that as n increases, the variance of $x_{1,n}$ grows too quickly. Using the parameters $\mu = 1$, $a = 1/20$, $b = 100/20$, already with $n = 10$ some of the values of the sample covariance are so large that computers treat them as infinite. Algorithms that rely on the sample covariance matrix or the Hessian to adaptively explore the target would not work properly in that case. This behavior onsets for even lower values of n if μ has a value far from zero, and a is small (with respect to the standard parametrization). Hence we recommend using the block structure, to be able to increase n at pleasure while avoiding uncontrolled behavior, which is particularly undesirable in a test problem.

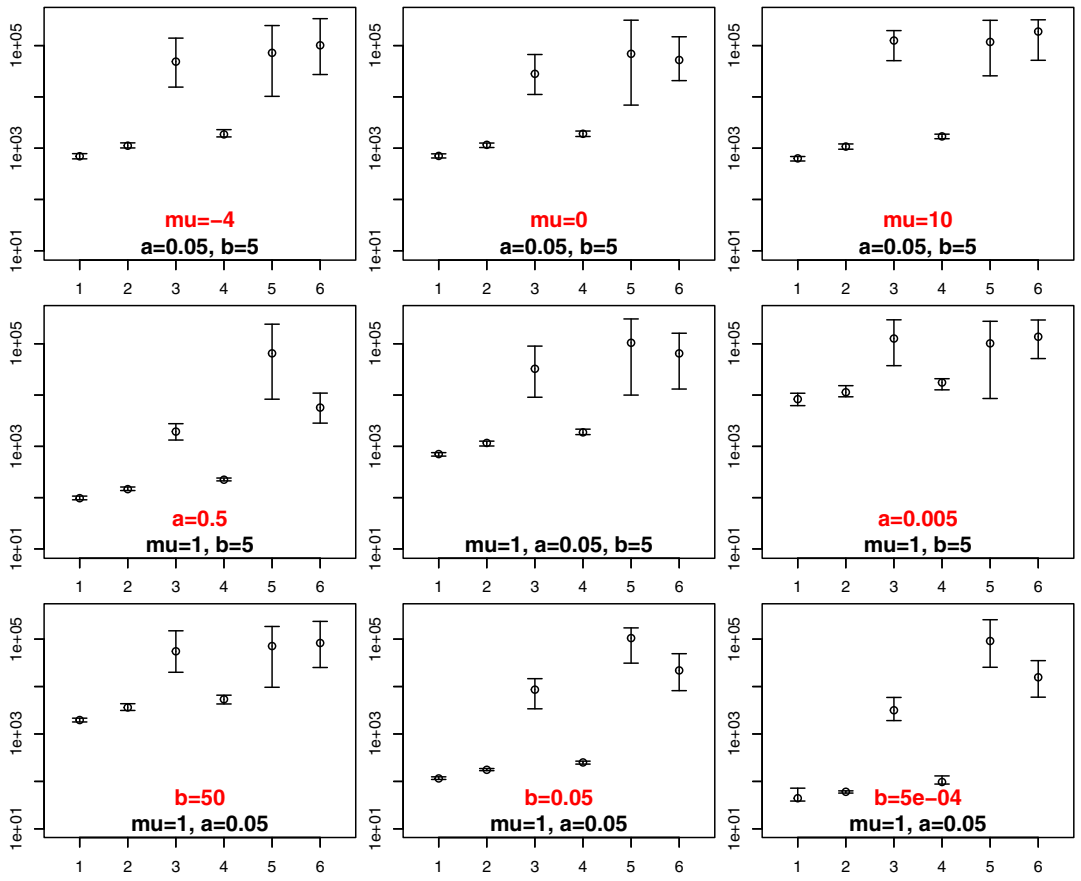

FIGURE 5 Integrated autocorrelation times τ obtained varying the parameters μ, a, b for Models 1 to 6, as described in Figure 4. The horizontal axis shows the single parameter value of either μ, a or b that takes a different value from the standard parametrization (i.e., $\mu = 1, a = 1/20$, and $b = 100/20$). The dot represents the average value, while the whiskers represent the two-sided 95% credibility region [Color figure can be viewed at wileyonlinelibrary.com]

As the choices for the shape parameters μ, a, and b are infinite, we performed our experiments for nine different sets of parameters, which provide a wide selection of the shapes that the Hybrid Rosenbrock distribution can take. For reference, Figure B1 in Appendix B shows how the shape of the first two variables of a Hybrid Rosenbrock vary for each parameter set we chose to utilize, complementing the description we gave in Section 3. From now on, we will refer to the set of parameters $\mu = 1, a = 1/20$, and $b = 100/20$ (values that originate from Equation 1) as the standard parametrization, which occupies the central plot in Figure 5.

Figure 5 shows how sMMALA tends to perform well (low τ) on Models 1, 2, and 4, with low variability, while it tends to perform worse on Models 3, 5, and 6, with higher variability. This behavior is explained by the fact that Models 1, 2, and 4 are all parametrized by the same value of $n_1 = 2$, while significant differences in the scales of the components start appearing only when $n_1 \geq 3$, as is the case for Models 3, 5, and 6. Therefore the difference between the variance of the components of the state space appears to be responsible for the increased difficulty of sampling from Models 3, 5, and 6. However, there does not appear to be a significant difference between Models 3, 6, both with a value of $n_1 = 3$, and Model 5, with a value of $n_1 = 5$. This suggests that

increasing the value of n_1 beyond three creates difficulties even for a sophisticated algorithm like sMMALA.

Moreover, examining the performance of sMMALA in Figure 5 in conjunction with the distribution shapes (Figure B1) suggests that models with values of μ, a, and b that yield rounder and more concentrated shapes—that is, large values of a, small values of b and to a lesser extent, values of μ near zero—tend to have lower τ values. Conversely, values of μ, a, and b that yield narrower and more elongated shapes—that is, large a, small b and μ far from zero—tend to have higher τ values.

Notably, the integrated autocorrelation time from Model 5 does not seem to be sensitive to changes in the shape parameters μ, a, and b. However, the uncertainty is quite large, so more computationally intensive tests may be needed to pinpoint the exact effects that the parameters μ, a, and b have on the difficulty of sampling from Model 5 with a sMMALA algorithm. It is likely that the tails of Model 5 are so elongated that even a state of the art algorithm like sMMALA has difficulties exploring them sufficiently well. Nonetheless, as explained in Remark 2, such results may not be particularly useful in practice.

4.2 | Algorithm comparison

In this section we study how the performance of popular MCMC algorithms changes as we vary the parameters of our model. Preliminary experiments suggest that, due to the difficult nature of the target, the integrated autocorrelation time of simple or naively tuned algorithms may be too large to be measured accurately with our computational means. Results based on the ESS and ESS per second (ESS/s) are similarly affected[1].

In this section we compare MCMC algorithms in terms of both the Kolmogorov–Smirnov (KS) distance, and the Anderson–Darling (AD) distance. The KS distance for the ith component of the state space is defined as

$$D_i = \sup_{x_i} |\tilde{F}_i(x_i) - F_i(x_i)|, \quad i = 1, \dots, n,$$

that is, the supremum of the absolute value of the difference between the empirical cumulative distribution function (CDF) constructed from the MCMC sample, $\tilde{F}_i$, and the empirical CDF constructed from a large *i.i.d.* Monte Carlo sample, F_i. We then only store the largest KS distance on the marginal distributions, which is simply

$$D = \max_{i=1,\dots,n} D_i.$$

The KS distance is a good measure of similarity between two CDFs. However, as the supremum of the distance between two CDFs is quite sensitive to local variation, it sometimes focuses on the region around the mode while ignoring differences in the tails. To address this limitation, we also measured the AD distance, which is defined as

$$A_i^2 = N \int_{-\infty}^{+\infty} \frac{(\tilde{F}_i(x_i) - F_i(x_i))^2}{F_i(x_i)(1 - F_i(x_i))} \, dF_i(x_i), \quad i = 1, \dots, n,$$

[1]For example, in Appendix E we show how, on Model 6 with standard parametrization, 10^8 samples from a RWM and a naively tuned HMC produce QQ-plots that are worse than those from 10^6 samples from the sMMALA algorithm.

where N is the number of samples, and only stored the largest distance among all components,

$$A^2 = \max_{i=1,\dots,n} A_i^2.$$

The AD distance is a squared distance similar to the Cramer–Von Mises (CM) distance. Like the CM distance, the integrand consists of the square difference between two CDFs, which is then weighted by the term $(F_i(x_i)(1-F_i(x_i)))^{-1}$. The AD distance tends to assign more weight to the tails of the distribution than the CM distance or the KS distance (Stephens, 1974), which is where we expect to find discrepancies in our analysis.

In practice, we constructed F_i from 100 million *i.i.d.* Monte Carlo samples, which is a considerable number for our purposes. Every marginal CDF F_i constructed this way takes up 2 Gb of memory on our machine, and our tests suggest that the effect of the sample variation is orders of magnitude smaller than the results we obtain in our experiments.

In this section we restrict our attention to Model 6, and we vary the parameters to change the difficulty of sampling from it via MCMC. In particular, we focus on the parameter b, which controls the dispersion of the distribution around the curved ridge. Roughly, the smaller the value of b, the flatter the distribution becomes, thereby making it easier for the MCMC algorithms to jump from one arm of the distribution to the other.

The MCMC algorithms tested in this section are the Random Walk Metropolis (RWM) (Metropolis et al., 1953), Hamiltonian Monte Carlo (HMC) (Duane et al., 1987; Neal, 2010), simplified Manifold MALA (sMMALA) (Girolami et al., 2011) with SoftAbs metric (Betancourt, 2013), and NUTS (Homan & Gelman, 2014) as implemented in (Carpenter et al., 2017). An important difference that sets these algorithms apart is that RWM, HMC, and NUTS explore the target using a global metric (i.e., both the variance of the transition kernel and the mass matrix do not depend on the current position of the algorithm), while the sMMALA algorithm takes advantage of the local curvature of the target when proposing moves in the state space. This property makes sMMALA particularly well suited to sample from targets like the Hybrid Rosenbrock distribution, and it provides a meaningful term of comparison for the RWM and HMC algorithms in this setting. Even though the details of the sMMALA algorithm are not essential to the exposition of our findings, we included a description of the algorithm in Appendix D.

We tuned the RWM to accept about 23% of the proposed moves (Roberts & Rosenthal, 1997), although in practice this probability oscillated between 22% and 29% depending on how difficult the target was. We took the variance of the transition kernel to be the identity matrix, as the local correlation structure changes significantly depending on the current position of the algorithm.

Tuning HMC on our target is particularly difficult as it is characterized by strong nonlinear correlations, and the optimality results in Beskos et al. (2013) only apply to multivariate normal targets with independent components. We decided to tune HMC naively, ignoring the problems that a curved density may cause to MCMC algorithms, and choosing the parameter values that yielded the best ESS/s. We took the mass matrix to be the identity matrix. We maintained the acceptance ratio between 78% and 85%, and found that 20 leapfrog steps produced an acceptable result for all the values of b considered in this section. The step sizes used are shown in the below table.

b	0.0005	0.005	0.05	0.5	5	50
HMC step size	0.026	0.048	0.08	0.25	0.4	0.7

We tuned sMMALA to maximize the value of the KS and AD distance, which yielded a different acceptance probability for each value of b, as shown in the below table. We used a value of $\alpha = 10^6$ for the correction parameter (see Appendix D for the meaning of α).

b	0.0005	0.005	0.05	0.5	5	50
sMMALA acceptance	0.38	0.33	0.26	0.24	0.24	0.29

The algorithm NUTS was also tuned to obtain the best result in terms of the KS and the AD distance. Based on purely empirical arguments, we sought an acceptance of 98%. We selected a diagonal mass matrix, as that yields slightly more stable results than the identity matrix in this case.

The number of iterations for RWM, HMC, and sMMALA was selected by measuring the same wall clock time for the three algorithms, and making sure that the number of iterations was large enough to yield a good result, yet not too large for our computational resources. The results are 21.5 million samples for RWM, 1 million samples for HMC, and 1.2 million samples for sMMALA. Tuning NUTS in the same way was problematic, as our RMW, HMC, and sMMALA algorithms are written in R, while NUTS and the package Stan are written in C++, which is significantly faster. In order to balance our analysis, once the number of iterations had been selected for RWM, HMC, and sMMALA, we counted the number of gradient evaluation that our naive HMC performed during the whole run. We then run NUTS and stopped it when the number of leapfrog steps (i.e., gradient evaluations) reached the same value. A consequence of this setup is that the number of iterations that NUTS performed was different for every run.

Figure 6 shows the results of our analysis. We repeated the experiments 16 times in order to assess the uncertainty around our estimates. The dot represents the mean value of the KS and AD distance, while the whiskers represent the two-sided 95% credibility region around our estimate. Note that both axes are expressed in the logarithmic scale.

As mentioned before, low values of the parameter b make the distribution flatter around the ridge, and easier to sample from. High values of b make the distribution narrower, restricting movement and hampering exploration. The results from the sMMALA algorithm support this view, with low values of the KS and AD distance for small values of b, and large values of higher values of the KS and AD distance for large values of b.

Surprisingly, despite tuning RWM as optimally as possible and taking a large number of samples, the performance of the algorithm is quite poor, even for low values of b. This result highlights the risk of trusting simple MCMC methods when the target is higher dimensional and exhibits curved correlation structure, even though the number of samples drawn may be large.

Interestingly, the naively tuned HMC algorithm seems to achieve results that are only marginally better than RWM. The uncertainty bars on the KS distance between RWM and HMC overlap in most of the measurements. The naively tuned HMC seems to achieve better results in terms of AD distance, suggesting that it explores the tails better than RWM.

Perhaps the most interesting detail of Figure 6 is the performance of NUTS. For high values of b, it equals the performance of sMMALA in terms of KS distance, while it surpasses sMMALA for low values of b. In terms of AD distance, NUTS is superior to sMMALA for all values of b considered, as much as by two orders of magnitude. This result suggests that the coherent exploration of straight ridges that characterizes HMC is also effective on curved ridges, when the number of steps is adjusted properly. On the contrary, sMMALA explores the state space by taking one step at a time, which can lead to diffusive behavior and slower exploration. However, the high

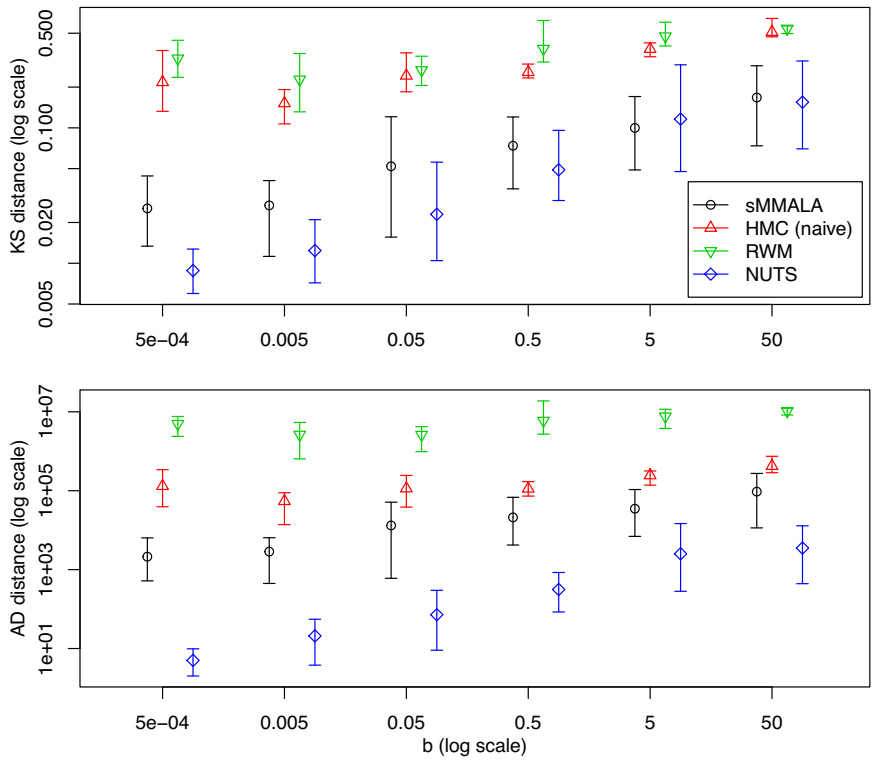

FIGURE 6 Performance comparison of the algorithms sMMALA, HMC, RWM, and NUTS in terms of the Kolmogorov–Smirnov distance, and of the Anderson–Darling distance, measured on Model 6 for different values of the parameter b. The dot represents the average value of 16 runs, while the whiskers represent the two-sided 95% confidence region around the estimate [Color figure can be viewed at wileyonlinelibrary.com]

computational cost of each NUTS iteration leads to a small final sample, which may be undesirable for certain applications. The algorithm sMMALA may be a more effective compromise between precision and final sample size.

5 | CONCLUSIONS

The 2-d Rosenbrock distribution and its current extensions to higher dimensions are common benchmark models in the field of Markov chain Monte Carlo. However, their normalization constant is generally unknown, and their shape seems to change in unexpected ways as the dimension of the state space increases, as shown in Section 2. These features characterize them as poor benchmark models for assessing the quality of MCMC samples. This is particularly true in higher dimensions, when visual inspection involves a significant amount of resources and is not always possible. A poor benchmark model may cause confusion in the interpretation of the results, as it can appear that an algorithm is working appropriately, while it is struggling to explore entire regions of the support of the distribution.

In this article we present the Hybrid Rosenbrock distribution, a reliable benchmark model with curved correlation structure that addresses all the shortcomings of the Rosenbrock distribution and of its higher dimensional extensions. The Hybrid Rosenbrock has a very challenging structure, due to how the conditional normal kernels are linked to each other.

Its shape can be made arbitrarily easier or harder to sample from by varying its parameters, for which we provide clear guidance. In Section 3 we give its normalization constant, and an algorithm to obtain an *i.i.d.* Monte Carlo sample without having to rely on MCMC samplers. The ability to sample the model directly proved to be very useful in Section 4.2, where we tested and compared the performance of some popular MCMC algorithms when sampling from the Hybrid Rosenbrock distribution.

Our results from Section 4 show how popular MCMC algorithms are not equipped to sample from a target with curved correlation structure, especially in high dimensions, and perform poorly. Despite that, common performance metrics used by practitioners may still suggest that the algorithms are performing well. A sophisticated algorithm such as sMMALA or NUTS can help detect the problem, but models used in the applied sciences are often too complex to obtain all the quantities sMMALA needs (e.g., the gradient and Hessian matrix of the target). This highlights the importance of testing MCMC algorithms on a reliable benchmark problem such as the Hybrid Rosenbrock, where results can be easily compared against the ground truth.

ACKNOWLEDGMENTS

This research was funded by the UK Medical Research Council programme MC_UU_00002/10 (Filippo Pagani, Martin Wiegand), the Engineering and Physical Sciences Research Council EP/R018561/1 (Filippo Pagani), and the University of Manchester (Filippo Pagani, Saralees Nadarajah). Filippo Pagani would like to thank Dr. Tim Waite, Dr. Simon Cotter, the Reviewers, and the Editors for their useful comments.

CONFLICT OF INTEREST

The authors have no conflict of interest to declare.

ORCID

Filippo Pagani https://orcid.org/0000-0003-1584-0881
Saralees Nadarajah https://orcid.org/0000-0002-0481-0372

REFERENCES

Allanach, B. C., & Lester, C. G. (2008). Sampling using a 'bank' of clues. *Computer Physics Communications*, *179*, 256–266.

Beskos, A., Pillai, N., Roberts, G., Sanz-Serna, J.-M., & Stuart, A. (2013). Optimal tuning of the hybrid Monte Carlo algorithm. *Bernoulli*, *19*, 1501–1534. https://doi.org/10.3150/12-BEJ414

Betancourt, M. (2013). *A general metric for Riemannian manifold Hamiltonian Monte Carlo*. In F. Nielsen & F. Barbaresco (Eds.), *Geometric science of information* (pp. 327–334). Springer.

Chib, S. (1995). Marginal likelihood from the Gibbs output. *Journal of the American Statistical Association*, *90*, 1313–1321.

Christen, J. A., & Fox, C. (2010). A general purpose sampling algorithm for continuous distributions (the t -walk). *Bayesian Analysis*, *5*, 263–281. https://doi.org/10.1214/10-BA603

Christensen, O. F., Roberts, G. O., & Rosenthal, J. S. (2005). Scaling limits for the transient phase of local metropolis–Hastings algorithms. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *67*, 253–268.

Cotter, C., Cotter, S., & Russell, P. (2019). Ensemble transport adaptive importance sampling. *SIAM/ASA Journal on Uncertainty Quantification*, *7*, 444–471. https://doi.org/10.1137/17M1114867

DiCiccio, T. J., Kass, R. E., Raftery, A., & Wasserman, L. (1997). Computing Bayes factors by combining simulation and asymptotic approximations. *Journal of the American Statistical Association*, *92*, 903–915.

Dixon, L., & Mills, D. (1994). Effect of rounding errors on the variable metric method. *The Journal of Optimization Theory and Applications*, *80*, 175–179.

Duane, S., Kennedy, A., Pendleton, B., & Roweth, D. (1987). Hybrid Monte Carlo. *Physics Letters B*, *195*, 216–222.

Feroz, F., Hobson, M., & Bridges, M. (2009). MultiNest: An efficient and robust Bayesian inference tool for cosmology and particle physics. *Monthly Notices of the Royal Astronomical Society*, *398*, 1601–1614.

Gelfand, A. E., & Smith, A. F. M. (1990). Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association*, *85*, 398–409.

Geyer, C. J. (1992). Practical Markov Chain Monte Carlo. *Statistical Science*, *7*, 473–483. https://doi.org/10.1214/ss/1177011137

Girolami, M., Calderhead, B., & Chin, S. (2011). Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *Journal of the Royal Statistical Society, Series B (Methodological)*, *73*(2), 123-214.

Goodman, J., & Weare, J. (2010). Ensemble samplers with affine invariance. *Communications in Applied Mathematics and Computational Science*, *5*, 65–80.

Hogg, D. W., & Foreman-Mackey, D. (2018). Data analysis recipes: Using Markov Chain Monte Carlo. *The Astrophysical Journal Supplement Series*, *236*, 11.

Homan, M. D., & Gelman, A. (2014). The no-u-turn sampler. *Journal of Machine Learning Research*, *15*, 1351–1381.

House, T., Ford, A., Lan, S., Bilson, S., Buckingham-Jeffery, E., & Girolami, M. (2016). Bayesian uncertainty quantification for transmissibility of influenza, norovirus and Ebola using information geometry. *Journal of the Royal Society Interface*, *13*, 20160279.

Kim, S., Shephard, N., & Chib, S. (1998). Stochastic volatility: Likelihood inference and comparison with ARCH models. *The Review of Economic Studies*, *65*, 361–393.

Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., & Teller, E. (1953). Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, *21*, 1087–1092.

Moral, P. D., Doucet, A., & Jasra, A. (2006). Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, *68*, 411–436.

Neal, R. (2010). MCMC using Hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, *54*, 113–162.

Parno, M. (2014) *Transport Maps for Accelerated Bayesian Inference* (Ph.D. thesis). MIT Computational Science and Engineering.

Roberts, G., & Rosenthal, J. (1997). Optimal scaling of discrete approximations to Langevin diffusions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *60*, 255–268.

Roberts, G. O., & Rosenthal, J. S. (2001). Optimal scaling for various metropolis-Hastings algorithms. *Statistical Science*, *16*, 351–367. https://doi.org/10.1214/ss/1015346320

Rockwood, L. L. (2015). *Introduction to population ecology* (2nd ed.). Wiley-Blackwell.

Rosenbrock, H. H. (1960). An automatic method for finding the greatest or least value of a function. *The Computer Journal*, *3*, 175–184.

Satagopan, J. M., Newton, M. A., & Raftery, A. E. (2000). *Easy estimation of normalizing constants and bayes factors from posterior simulation: Stabilizing the harmonic mean estimator. Technical report*. University of Washington.

Carpenter, B., Gelman, A., Hoffman, M.D., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M., Guo J., Li,P. & Riddell, A. (2017). Stan: A Probabilistic Programming Language. *Journal of Statistical Software*, *76*, (1). http://dx.doi.org/10.18637/jss.v076.i01.

Stephens, M. A. (1974). EDF statistics for goodness of fit and some comparisons. *Journal of the American Statistical Association*, *69*, 730–737. Retrieved from. http://www.jstor.org/stable/2286009

Sullivan, A. B., Snyder, D. M., & Rounds, S. A. (2010). Controls on biochemical oxygen demand in the upper Klamath River, Oregon. *Chemical Geology*, *269*, 12–21 Rates of Geochemical Processes and their Application to Natural Systems.

The Dark Energy Survey Collaboration (2017) Cosmology from cosmic shear with DES science verification data. *arXiv*. Retrieved from https://arxiv.org/abs/1507.05552

Thompson, M. B. (2010) A comparison of methods for computing autocorrelation time. *arXiv preprint arXiv:1011.017*.

How to cite this article: Pagani F, Wiegand M, Nadarajah S. An *n*-dimensional Rosenbrock distribution for Markov chain Monte Carlo testing. *Scand J Statist*. 2021;1–24. https://doi.org/10.1111/sjos.12532

APPENDIX A. CURRENT LITERATURE

In three dimensions the Full Rosenbrock kernel in Equation (2) can be written as

$$\pi(\mathbf{x}) \propto \exp\left\{-\left[100\,(x_2 - x_1^2)^2 + (1-x_1)^2 + 100\,(x_3 - x_2^2)^2 + (1-x_2)^2\right]/20\right\} \qquad \mathbf{x} \in \mathbb{R}^3. \tag{A1}$$

Figure A1 shows contour plots of a 2 million sample obtained running a sMMALA algorithm (described in Appendix D) on Equation (A1), with starting point $\mathbf{x} = [1, \ldots, 1]^\top$, $\alpha = 10^6$, and an acceptance ratio of about 50%.

Note the difference in curvature between the (x_1, x_2) plot and the (x_1, x_3) plot. Also, the three variables involved have very different variances, as can be seen from the scales of the plots.

Figure A2 shows the shape of the 2-d marginal distributions of (3) when taking $n = 4$ and $\mu_1 = \mu_3 = 1$, which result in the kernel

$$\pi(\mathbf{x}) \propto \exp\left\{-\left[(x_1 - 1)^2 + 100\,(x_2 - x_1^2)^2 + (x_3 - 1)^2 + 100\,(x_4 - x_3^2)^2\right]/20\right\}. \tag{A2}$$

The contours were plotted using a sample from a sMMALA algorithm tuned exactly as described in the previous section, with $\alpha = 10^6$, $\mathbf{x} = \underline{1}$, and acceptance ratio roughly 50%.

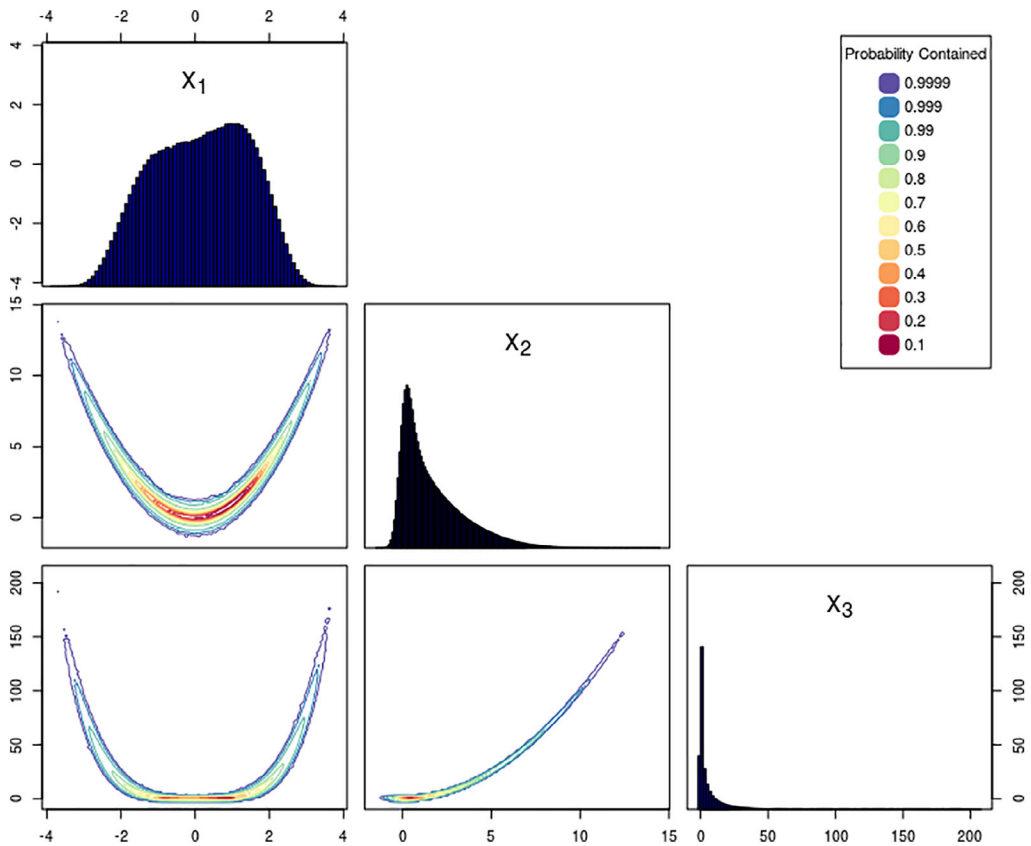

FIGURE A1 Contour plot of a 3-d Full Rosenbrock density, as described in Equation (A1), obtained from a sMMALA MCMC sample [Color figure can be viewed at wileyonlinelibrary.com]

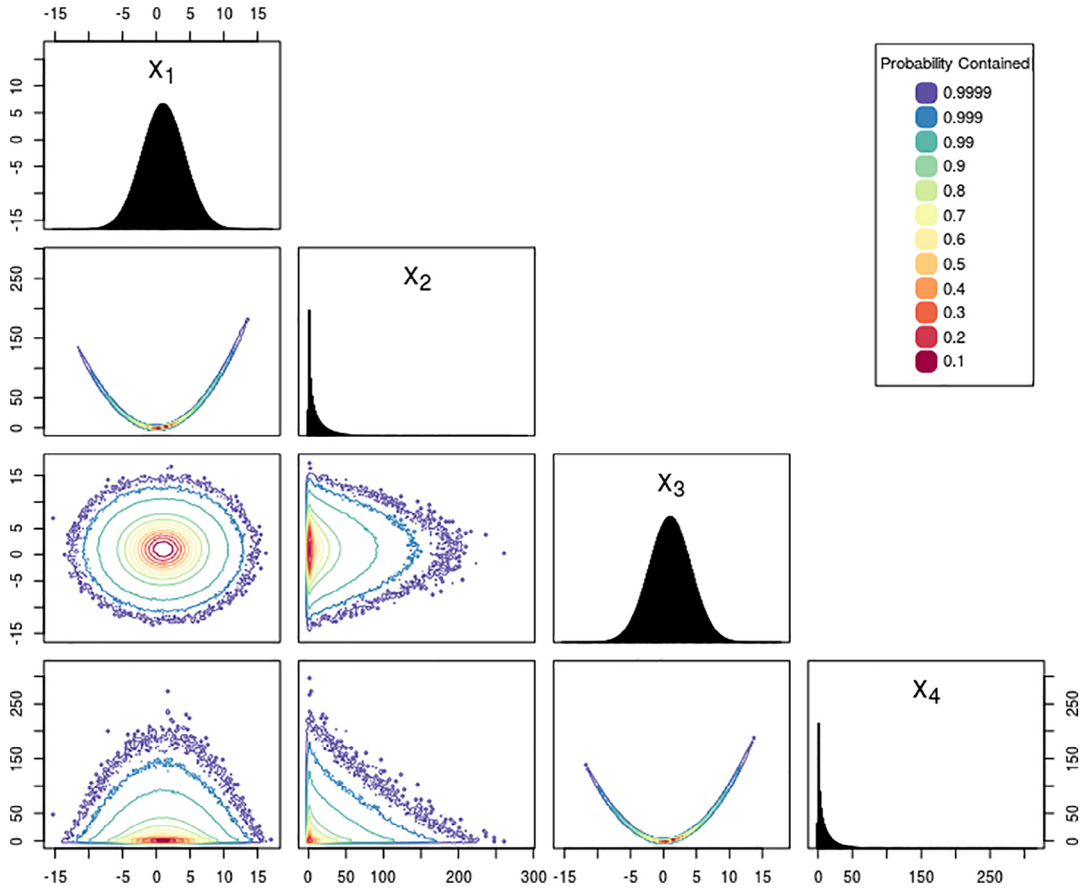

FIGURE A2 Contour plot of a 4-*d* Even Rosenbrock density, as described in Equation (3). Most of the joint distributions are uncorrelated [Color figure can be viewed at wileyonlinelibrary.com]

Equation (3) represents a more straightforward problem than Equation (A1). The round shapes and lack of ridges in the lower left plots of Figure A2 (specifically for the pairs $(x_1, x_3), (x_1, x_4), (x_2, x_3)$, and (x_2, x_4)) confirm the lack of complex dependencies that characterize the Full Rosenbrock kernel.

APPENDIX B. INTERPRETATION OF THE PARAMETERS

We can rewrite the 2-*d* Rosenbrock kernel from Equation (1) in general form as:

$$\begin{aligned}\pi(x_1, x_2) &\propto \exp\left\{-a(x_1-\mu)^2 - b(x_2 - x_1^2)^2\right\} \\ &\propto \exp\left\{-\frac{1}{2\frac{1}{2a}}(x_1-\mu)^2 - \frac{1}{2\frac{1}{2b}}(x_2 - x_1^2)^2\right\}. \end{aligned} \tag{B1}$$

where $a = 1/20$, $b = 100/20$, and $\mu = 1$, and more generally $a, b \in \mathbb{R}^+, \mu \in \mathbb{R}$, $x_1, x_2 \in \mathbb{R}$. Equation (B1) should make it clear that the density is composed of two normal kernels, that is,

$\pi(x_1, x_2) = \pi(x_1)\pi(x_2|x_1)$, where

$$\pi(x_1) \sim \mathcal{N}\left(\mu, \frac{1}{2a}\right), \quad \pi(x_2|x_1) \sim \mathcal{N}\left(x_1^2, \frac{1}{2b}\right).$$

This allows us to calculate the normalization constant as follows.

Proposition 2. *The normalization constant of the 2-d Rosenbrock kernel as shown in Equation (B1) is* $\sqrt{ab}/\pi$.

Proof. We begin by integrating Equation (B1) over the domain $\mathbb{R}^2$:

$$\begin{aligned}
&\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \exp\left\{-\frac{1}{2\frac{1}{2a}}(x_1-\mu)^2 - \frac{1}{2\frac{1}{2b}}(x_2-x_1^2)^2\right\} dx_2\, dx_1 \\
&= \int_{-\infty}^{\infty} \exp\left\{-\frac{1}{2\frac{1}{2a}}(x_1-\mu)^2\right\} \int_{-\infty}^{\infty} \exp\left\{-\frac{1}{2\frac{1}{2b}}(x_2-x_1^2)^2\right\} dx_2\, dx_1.
\end{aligned}$$

We can apply a change of variables in the second integral, $v = x_2 - x_1^2$, which becomes

$$= \int_{-\infty}^{\infty} \exp\left\{-\frac{1}{2\frac{1}{2a}}(x_1-\mu)^2\right\} \int_{-\infty}^{\infty} \exp\left\{-\frac{1}{2\frac{1}{2b}}v^2\right\} dv\, dx_1,$$

expression that highlights the two kernels $x_1 \sim \mathcal{N}(\mu, 1/2a)$ and $v \sim \mathcal{N}(0, 1/2b)$. Solving the integrals individually,

$$\begin{aligned}
&= \sqrt{2\pi\frac{1}{2a}}\,\sqrt{2\pi\frac{1}{2b}} \\
&= \frac{\pi}{\sqrt{ab}}.
\end{aligned}$$

The reciprocal of this number is the normalization constant. ∎

The normalization constant of the Hybrid Rosenbrock distribution can be calculated following similar steps, and can be found in Appendix C.

Figure B1 shows how changing the parameters μ, a, and b influences the shape of the Rosenbrock density, which coincide with the first two dimensions of the Hybrid Rosenbrock distribution. The central plot shows the shape of Equation (1), with parameters $\mu = 1$, $a = 1/20$, and $b_{1,2} = 100/20$, which we will refer to as the "standard parametrization." Every other plot shows notable shapes we were able to obtain by varying the parameters one at a time.

Remark 3. In light of this, we are able to explain why the Full Rosenbrock kernel changes shape as its dimension increases. For simplicity, we will illustrate our point using a 3-*d* Full Rosenbrock model. From Equation (A1) we can derive the following general expression:

$$\pi(\mathbf{x}) \propto \exp\left\{-a(x_1-\mu_1)^2 - b(x_2-x_1^2)^2 - c(x_2-\mu_2)^2 - d(x_3-x_2^2)^2\right\}, \tag{B2}$$

where $\mathbf{x} \in \mathbb{R}^3$ and $a, b, c, d \in \mathbb{R}^+$, $\mu_1, \mu_2 \in \mathbb{R}$. While the first and fourth terms are two normal kernels in x_1 and x_3 and can be easily isolated, the x_2 kernel is now composed of two terms.

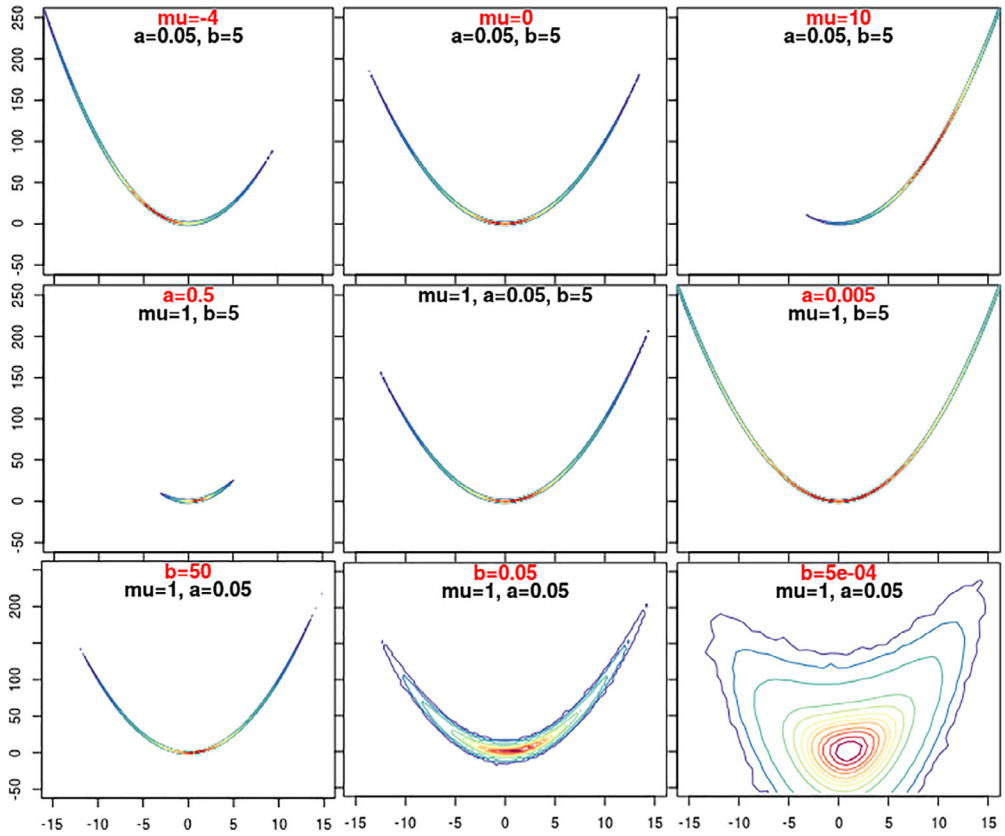

FIGURE B1 Contour plot for the variables $(x_1, x_{1,2})$ of a Hybrid Rosenbrock density, as the parameters μ, a, and b take different values. For comparison, the central plot represents a kernel with the original values of the parameters, that is, $\mu = 1$, $a = 1/20$, and $b_{1,2} = 100/20$ [Color figure can be viewed at wileyonlinelibrary.com]

Consequently, the integral of (B2) with respect to x_3, does not yield Equation (1). In order to obtain a more compact expression for the kernel in the variable x_2, we expand the second and third terms of Equation (B2) to a sum of monomials, and complete the squares by adding and subtracting the necessary terms:

$$\begin{aligned} &-b(x_2 - x_1^2)^2 - c(x_2 - \mu_2)^2 \\ &\quad = -\frac{1}{2}\left(\frac{x_2 - \frac{2bx_1^2 + 2c\mu_2}{2b+2c}}{\frac{1}{\sqrt{2b+2c}}}\right)^2 - \frac{(2bx_1^2 + 2c\mu_2)^2}{2(2b+2c)} - \frac{1}{2}(2bx_1^4 + 2c\mu_2^2), \end{aligned} \tag{B3}$$

which can be substituted back in (B2). The first term in (B3) represents the new normal kernel for x_2, that is,

$$x_2|x_1 \sim \mathcal{N}\left(\frac{(2bx_1^2 + 2c\mu_2)^2}{2b+2c}, \frac{1}{2b+2c}\right). \tag{B4}$$

This kernel is not influenced by the x_2 variable present in the last term of (B2) as it disappears after integrating in the variable x_3, as we showed in the proof of Proposition 2. The other terms in

(B3) are remaining terms from the calculations that we cannot simply include in the normalizing constant, as they depend on x_1. This changes the kernel of the variable x_1, whose distribution is now unknown and cannot be sampled from directly. The variable $x_2|x_1$ also changes shape drastically. Looking at Equation (B4), the value of the variance of $x_2|x_1$ changes from $1/2b$ to $1/(2b+2c)$, producing the effect observed in Figure 1.

These considerations extend to higher dimensions ($n>3$), where every time the dimension of the model is increased to $n+1$ according to the scheme in (2), the kernels of the variables $x_1, \dots, x_n$ change as described above.

APPENDIX C. DETAILS OF PROOF OF PROPOSITION 1

The integral of Equation (4) over the domain $\mathbb{R}^n$ is

$$\begin{aligned}\int_{\mathbb{R}^n} \exp\left\{-a(x_1-\mu)^2 - \sum_{j=1}^{n_2}\sum_{i=2}^{n_1} b_{j,i}(x_{j,i}-x_{j,i-1}^2)^2\right\} dx_{n_2,n_1} \dots dx_1 \\ = \int_{\mathbb{R}} \exp\left\{-a(x_1-\mu)^2\right\} \prod_{j=1}^{n_2}\prod_{i=2}^{n_1} \int_{\mathbb{R}} \exp\left\{-b_{j,i}(x_{1,i}-x_{1,i-1}^2)^2\right\} dx_{n_2,n_1} \dots dx_1, \end{aligned} \tag{C1}$$

by splitting the terms in the exponential function. We can now isolate the last integral, with indices $j=n_2$ and $i=n_1$, as

$$\begin{aligned}= \int_{\mathbb{R}} \exp\left\{-a(x_1-\mu)^2\right\} \prod_{j=1}^{n_2}\prod_{i=2}^{n_1-1} \int_{\mathbb{R}} \exp\left\{-b_{j,i}(x_{1,i}-x_{1,i-1}^2)^2\right\} \\ \times \int_{\mathbb{R}} \exp\left\{-b_{n_2,n_1}(x_{n_2,n_1}-x_{n_2,n_1-1}^2)^2\right\} dx_{n_2,n_1} dx_{n_2,n_1-1} \dots dx_1. \end{aligned} \tag{C2}$$

From Proposition 2 we know that by changing variables $v_{n_2,n_1} = x_{n_2,n_1} - x_{n_2,n_1}^2$,

$$\int_{\mathbb{R}} \exp\left\{-b_{n_2,n_1}(x_{n_2,n_1}-x_{n_2,n_1-1}^2)^2\right\} dx_{n_2,n_1} = \sqrt{\frac{\pi}{b_{n_2,n_1}}}.$$

We can substitute this result back into (C2), which becomes

$$= \sqrt{\frac{\pi}{b_{n_2,n_1}}} \int_{\mathbb{R}} \exp\left\{-a(x_1-\mu)^2\right\} \prod_{j=1}^{n_2}\prod_{i=2}^{n_1-1} \int_{\mathbb{R}} \exp\left\{-b_{j,i}(x_{1,i}-x_{1,i-1}^2)^2\right\} dx_{n_2,n_1-1} \dots dx_1. \tag{C3}$$

We can apply the same procedure to all the integrals in Equation (C3) in turn, starting from the remaining last variable n_1-1 of the last block n_2, until the very first variable x_1. This operation yields

$$\int_{\mathbb{R}^n} \exp\left\{-a(x_1-\mu)^2 - \sum_{j=1}^{n_2}\sum_{i=2}^{n_1} b_{j,i}(x_{j,i}-x_{j,i-1}^2)^2\right\} dx_{n_2,n_1} \dots dx_1 = \frac{\pi^{n/2}}{\sqrt{a}\prod_{i=2,j=1}^{n_1,n_2}\sqrt{b_{j,i}}}.$$

Taking the reciprocal we obtain the normalization constant in the statement of Proposition 1.

APPENDIX D. THE SMMALA ALGORITHM

The sMMALA algorithm (Girolami et al., 2011), based on the MALA algorithm (Roberts & Rosenthal, 1997), is part of a class of methods that use local information about the target when proposing a move in the state space. The sMMALA algorithm will propose a new position $\mathbf{x}'$ in the state space from the current position $\mathbf{x}$ according to the equation

$$\mathbf{x}' = \mathbf{x} + \frac{h}{2}\Sigma(\mathbf{x})\nabla \log \pi(\mathbf{x}) + \mathcal{N}_n(0, h\ \Sigma(\mathbf{x})), \quad \mathbf{x} \in \mathbb{R}^n. \tag{D1}$$

Here $\pi(\mathbf{x})$ is the distribution of interest, ∇ represents the gradient operator, $\Sigma(\mathbf{x})$ is a positive definite matrix, and $h \in \mathbb{R}^+$ is the step size of the algorithm, parameter tuned by the user to achieve the desired level of acceptance. The proposed $\mathbf{x}'$ then is accepted with a Metropolis acceptance/rejection step, which ensures that the sMMALA sample comes from the correct stationary distribution $\pi(\mathbf{x})$.

A common choice of $\Sigma(\mathbf{x})$ is the Fisher Information matrix (i.e., the negative expectation of the Hessian of the log-likelihood; Girolami et al., 2011), as it carries information on the local correlation structure of the target. In our case the most convenient choice of $\Sigma(\mathbf{x})$ is given in Betancourt (2013), which uses a regularized version $\tilde{H}$ of the Hessian H of the log-density, derived by multiplying the eigenvectors of H by the absolute value of the eigenvalues. If the eigenvalues are too small, the eigendecomposition may be unstable, so the algorithm obtains $\tilde{H}$ by increasing the problematic eigenvalues by a factor of $1/\alpha$, where α is a user defined parameter.

APPENDIX E. SAMPLE BIAS

In this section we compared QQ-plots of MCMC samples for Model 6 with standard parametrization, that is, Equation (5) with $\mu = 1$, $a = 1/20$, and $b_{j,i} = 100/20$, $i = 2, 3$, $j = 1, 2, 3$. We selected this specific target as it is not overly challenging for our computational resources, but retains all the main features of the Hybrid Rosenbrock distribution: it is composed of multiple blocks with multiple variables per block.

We drew 10 million samples from the kernel in Equation (5) using Algorithm 1 to construct the quantiles of the marginal distribution. We then compared these quantiles with quantiles from 1 million sMMALA samples, 100 million RWM samples, and 100 million HMC samples. This is the largest sample size that we could feasibly analyze with our current computational means. Note how the number of RWM and HMC samples is significantly higher than what we previously used in our tests, while the number of sMMALA samples was left unchanged. However, the RWM and HMC sample bias remains high, as the QQ-plots for each variable in Equation (5) show in Figure E1.

The top left plot in Figure E1 shows the QQ-plot for the variable x_1. The black line, representing the empirical quantiles obtained from the sMMALA sample, remains reasonably close to the blue line, which represents the empirical quantiles calculated from Algorithm 1. Large discrepancies between the black and blue lines only occur far in the tails region. This is due to most MCMC algorithms having difficulties visiting the tails and returning to the mode efficiently, while direct Monte Carlo sampling does not suffer from this drawback. Repeating the same tests with a larger sMMALA sample provides empirical quantiles that match more closely to those constructed in this example, suggesting that sMMALA mixes reasonably well.

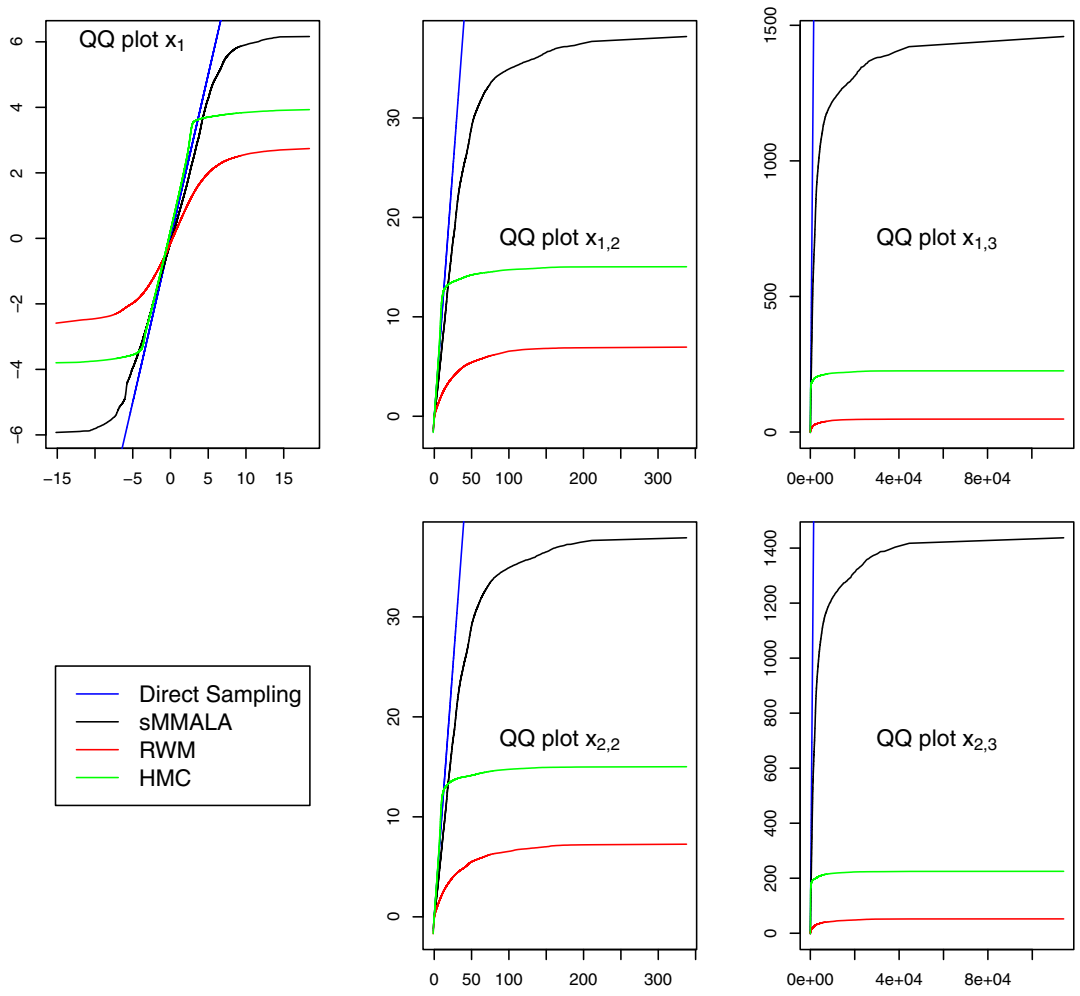

FIGURE E1 QQ-plots for each variable of the Hybrid Rosenbrock distribution (Equation 5). The horizontal axis show the quantiles obtained from direct Monte Carlo sampling, while the vertical axis shows the quantiles calculated from 100 million RWM MCMC samples, 100 million HMC, and 1 million sMMALA samples [Color figure can be viewed at wileyonlinelibrary.com]

On the other hand, the quantiles from RWM and HMC appear to be very far from the true solution, even despite the use of a considerably larger sample than in Section 4.2. Both algorithms appear to have troubles exploring the tails in both arms of the distribution, especially in the right arm, where the tail reaches farther. Despite the large number of samples, RWM never abandons the local region around the mode of the distribution. This is particularly important when trying to estimate tail probabilities in order to produce constraints on parameters, for example, in cosmology. HMC appears to be marginally better than RWM, but it should be noted that the tuning parameters used in this example make each HMC iteration between 15 and 20 times more expensive than a RWM iteration in terms of run time.

The variables $x_{1,2}$ and $x_{2,2}$, shown in the middle plots, have tails that stretch moderately far from the mode. Again, the algorithm sMMALA agrees quite well with the sample from Algorithm

1. However, the samples from RWM and HMC reveal how these algorithms struggle to abandon the mode.

The last two plots on the right side of Figure E1 show the QQ-plots for variables $x_{2,2}$ and $x_{2,3}$, which have tails that reach very far from the mode. Once again, the results from sMMALA and Algorithm 1 are in good agreement, while RWM and HMC never explore the tails.

In order to control for Monte Carlo error in the quantiles constructed using Algorithm 1, we repeated the experiment in this section with 20 million samples instead of 10. The results in Figure E1 did not change significantly, leading us to believe that the Monte Carlo error that Algorithm 1 introduces in our analysis is negligible.

These experiments show how despite taking very large MCMC samples and using well established metrics, having a reliable benchmark model is crucial when testing MCMC algorithms on difficult targets with curved correlation structure.